Semantic Nets are in the Eye of the Beholder

Lenhart K. Schubert

Technical Report 346
May 1990

UNIVERSITY OF
ROCHESTER
COMPUTER SCIENCE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 346	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Semantic Nets are in the Eye of the Beholder		5. TYPE OF REPORT & PERIOD COVERED technical report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lenhart K. Schubert		8. CONTRACT OR GRANT NUMBER(s) N00014-82-K-0193
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Dept., 734 Computer Studies Bldg. University of Rochester, Rochester, NY 14627		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd., Arlington, VA 22209		12. REPORT DATE May 1990
		13. NUMBER OF PAGES 13
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Inf. Systems, Arlington, VA 22217		15. SECURITY CLASS. (of this report) unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

semantic nets; knowledge representation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(see reverse)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT

The term "semantic nets," in its broadest sense, has become virtually meaningless. It is applied to systems which, as a class, lack distinctive representational and computational properties vis a vis other knowledge representation (KR) schemes. This terminological problem is not due to lack of substance or coherence of work done under the semantic net banner. Rather, it is due to convergence of the major KR schemes: the representational and computational strategies employed in semantic net systems are abstractly equivalent to those employed in virtually *all* state-of-the-art systems incorporating a substantial propositional knowledge base, whether they are described as logic-based, frame-based, rule-based, or something else. In particular, I will argue that using a graph-theoretic propositional representation does not automatically distinguish it from others: even sets of PC formulas, abstractly viewed, are graphs. Nor is "proximity-based" inference (using graph-theoretic distance) automatically distinctive, since even resolution strategies (with reasonable indexing schemes) are proximity-based in the abstract; nor is hierarchic property inheritance any longer distinctive, given its availability in state-of-the-art logic-based, frame-based, and rule-based systems. So I urge some more restrictive, and hence more meaningful use of the term "semantic nets" than is the current practice.

Semantic Nets are in the Eye of the Beholder

Lenhart K. Schubert

Department of Computer Science
University of Rochester
Rochester, NY 14627

May 17, 1990

Abstract. The term "semantic nets", in its broadest sense, has become virtually meaningless. It is applied to systems which, as a class, lack distinctive representational and computational properties *vis a vis* other knowledge representation (KR) schemes. This terminological problem is not due to lack of substance or coherence of work done under the semantic net banner. Rather, it is due to convergence of the major KR schemes: the representational and computational strategies employed in semantic net systems are abstractly equivalent to those employed in virtually *all* state-of-the-art systems incorporating a substantial propositional knowledge base, whether they are described as logic-based, frame-based, rule-based, or something else. In particular, I will argue that using a graph-theoretic propositional representation does not automatically distinguish it from others: even sets of PC formulas, abstractly viewed, are graphs. Nor is "proximity-based" inference (using graph-theoretic distance) automatically distinctive, since even resolution strategies (with reasonable indexing schemes) are proximity-based in the abstract; nor is hierarchic property inheritance any longer distinctive, given its availability in state-of-the-art logic-based, frame-based, and rule-based systems. So I urge some more restrictive, and hence more meaningful use of the term "semantic nets" than is the current practice.

1 Introduction

In AI, as in any science, ideas do not evolve linearly. There is no universally shared terminology, let alone a universally assimilated body of knowledge, which is augmented step-by-step by the "advances" we (as practitioners) announce. Rather, ideas tend to emerge and evolve in variant forms in many places, in more or less parallel fashion.

On the face of it, this parallelism is wasteful, but it has compelling causes. For one thing, too much is being written for any of us to read, so naturally we fragment into colonies that are internally cohesive, but only loosely integrated with each other. It is often faster to rediscover something within the framework of one's colony than to glean it from the writings of another. For another, our numbers are legion and our sources of inspiration widely shared, so if a good idea occurs to one of us, it is apt to occur to many; since ideas are our livelihood, we are disposed to emphasize differences – however superficial – rather than similarities, and jealously defend our terminological niches. Besides, we may argue, evolution thrives on diversity.

However, in the evolution of a set of related scientific notions, there is a time to explore diverse alternatives, and later, a time to consolidate ideas, discarding artificial distinctions and inconsistent terminology. It seems to me that the time for such a consolidation and redefinition has arrived in the area of KR. In particular, I want to argue that the term "semantic nets" has become virtually meaningless, at least in its broadest sense. (By the same token, questions can also be raised about "frame-based sytems", "rule-based systems", and various other KR terms, but I would like to focus on what I know best.) It no longer signifies an objectively distinguishable species in the KR taxonomy. *All* KR schemes I have lately encountered, which aspire to cope with a large, general propositional knowledge base, qualify as semantic nets, appropriately viewed. However, their designers often don't view them that way, preferring such terms as "frame-based system", "semantic database", "blackboard", or some neologism. The choice of KR terminology seems to be more a matter of intellectual affiliation, than one of substance.

More specifically, I will argue that semantic nets fail to be distinctive in the way they (1) represent propositions, (2) cluster information for access, (3) handle property inheritance, and (4) handle general inference; in other words, they lack distinctive representational properties (i.e., 1) and distinctive computational properties (i.e., 2-4). Certain propagation mechanisms, notably "spreading activation", "intersection search", or "inference propagation" have sometimes been regarded as earmarks of semantic nets, but since most extant semantic nets lack such mechanisms, they cannot be considered criterial in current usage. One way of re-invigorating the term, I will suggest, would be to restrict it to just such active networks. Another would be to reserve it for certain specialized representations (such as taxonomic and temporal graphs) which use graph-theoretic notions in an essential, nontrivial way.

2 Representational Properties of Semantic Nets

By the representational, as opposed to computational, properties of semantic nets I mean those aspects of their structure which are interpretable as *denoting* something in the domain: individuals, properties, relations, magnitudes, facts, states of affairs, and so on. With respect to these representational properties, semantic nets have often been called notational variants of logic (or rather, of various logics). In a certain sense, I concur; but what exactly does this mean?

Let us note at once that the term "notational" is prejudicial: it suggests that we are concerned with notations on paper (or display screens), whereas my concern here is with information structures in computers. These two notions of representation *are* related; for instance, sets of predicate logic expressions on paper and node-and-link diagrams (of the right type) would appear to be expressively equivalent just in case their corresponding computer realizations are expressively equivalent. Furthermore, it is hard to *write* about propositional representations in computers without resorting to some notation like bracketed expressions or diagrams on paper. However, if we blur the distinction between these two notions of representation, we are apt to get tangled up in questions that are not at issue here: for instance, whether node-and-link drawings provide more perspicuous propositional representations than bracketed expressions.

So in what follows, all references to propositional representations or "notations" are to be understood as references to information structures in computers. This does not mean, of course, that we are concerned with machine-oriented, hardware-level information structures, but rather with structures at some appropriate level of abstraction, such as the level of symbolic expressions or graphs.

A propositioal representation (in computers as well as on paper) has two parts: a syntax, and a semantics. (There is usually an associated *calculus* as well, for establishing new propositions from given ones, but that is a computational matter.) Suppose that two representations admit transformations from each to the other, such that any syntactically well-formed knowledge set (for lack of a better word) in one representation is effectively mapped into a well-formed knowledge set in the other, and furthermore, the original knowledge set and its transform are semantically equivalent. (Accept for the moment that this last notion can be made precise – I will elaborate shortly.) Then we have two effectively interchangable representations, enabling us to express exactly the same things. Can we therefore regard them as "notational" variants (more exactly, structural variants)?

It may be felt that this is not quite enough. After all, there seems to be a sentations (such as taxonomic and temporal graphs) which use graph-theoretic notions in an essential, nontrivial way.

2 Representational Properties of Semantic Nets

By the representational, as opposed to computational, properties of semantic nets, I mean those aspects of their structure which are interpretable as *denoting* something in the domain: individuals, properties, relations, magnitudes, facts, states of affairs, and so on. With respect to these representational properties, semantic nets have often been called notational variants of logic (or rather, of various logics). In a certain sense, I concur; but what exactly does this mean?

Let us note at once that the term "notational" is prejudicial: it suggests that we are concerned with notations on paper (or display screens), whereas my concern here is with information structures in computers. These two notions of representation *are* related; for instance, sets of predicate logic expressions on paper and node-and-link diagrams (of the right type) would appear to be expressively equivalent just in case their corresponding computer realizations are expressively equivalent. Furthermore, it is hard to *write* about propositional representations in computers without resorting to some notation like bracketed expressions or diagrams on paper. However, if we blur the distinction between these two notions of representation, we are apt to get tangled up in questions that are not at issue here: for instance, whether node-and-link drawings provide more perspicuous propositional representations than bracketed expressions.

So in what follows, all references to propositional representations or "notations" are to be understood as references to information structures in computers. This does not mean, of course, that we are concerned with machine-oriented, hardware-level information structures, but rather with structures at some appropriate level of abstraction, such as the level of symbolic expressions or graphs.

A propositioal representation (in computers as well as on paper) has two parts: a syntax, and a semantics. (There is usually an associated *calculus* as well, for establishing new propositions from given ones, but that is a computational matter.) Suppose that two representations admit transformations from each to the other, such that any syntactically well-formed knowledge set (for lack of a better word) in one representation is effectively mapped into a well-formed knowledge set in the other, and furthermore, the original knowledge set and its transform are semantically equivalent. (Accept for the moment that this last notion can be made precise – I will elaborate shortly.) Then we have two effectively interchangable representations, enabling us to express exactly the same things. Can we therefore regard them as "notational" variants (more exactly, structural variants)?

It may be felt that this is not quite enough. After all, there seems to be a similar correspondence between, say, programs of a universal Turing machine (i.e., tape expressions describing some other, or the same, Turing machine) and Lisp programs: there are effective (in fact, primitive recursive) mappings from each to the other, such that each program and its transform have exactly the same partial recursive function as extension (or the same class of partial recursive functions, if we allow for different ways of interpreting inputs and outputs as numbers or other abstract objects.) Yet the two representations may be so wildly different, and so lacking in any *structural* resemblance, that we would hesitate to call them mere "notational variants" of each other. (When we talk about a notational variant of Lisp, we might be thinking of something like Interlisp, but hardly universal Turing machine programs!)

However, there's more to the transformations from nets to sets of formulas, and the reverse, than mere effectiveness: they can be chosen to be *isomorphisms* with certain special properties. These special properties pertain to the *meaningful expressions* (meaningful parts) making up two corresponding knowledge sets. Specifically, such an isomorphism maps the atomic and compound expressions of one formalism to those of the other in a way that (i) associates with each interpretable atomic expression of one knowledge set an interpretable atomic expression of the transformed knowledge set which admits exactly the *same* interpretations (such as individuals in the domain of discourse, sets or relations over such individuals, intensions, properties, or what have you); and (ii) preserves semantically significant structural features of compound expressions (such as the subexpression relation, and subexpression ordering).

Such an isomorphism induces a very strong semantic correspondence: the two knowledge sets have the same "valuations", and hence the same models. By a valuation of a knowledge set I mean an assignment of semantic values to all of its meaningful expressions (parts), beginning with admissible interpretations of its interpretable atomic expressions and continuing with values for compound expressions determined in accordance with the semantic rules of the formalism (in logic, the rules for interpreting functional expressions, atomic formulas, logically compound formulas, and quantified formulas). A model of a knowledge set is a valuation that renders its "top-level" proposition-denoting expressions true.

Let me illustrate. Besides clarifying some of the preceding notions and claims, the illustration will serve to allay the following sort of skepticism, which experienced network theorists may by now feel: how can there be an *isomorphism* between nets and sets of formulas, when one of the clearest intuitions about nets is that they introduce just *one* node for each distinct entity, whereas sets of formulas duplicate individual tokens with prodigal abandon? The answer contained in the illustration is that we need not view sets of formulas in terms of the distinct tokens they contain, but rather can view them in terms of the distinct *expressions* they contain, however many occurrences (tokens) of those expressions there may be.

The illustration may strike the reader as a sort of extended pun – and so it is. I will define a particular, "bare-bones" type of semantic net called an *s-net*, serving as a propositional representation but devoid of any specific access structures or other computational features. This type of net is so closely related to logic that I will not need to construct an explicit isomorphism from one to the other. In fact it *is* logic (FOPC to be exact), but couched in terms that make it appear to be a net (a directed, acyclic, labelled graph). Thus from a purely representational perspective, formulas – appropriately viewed – *are* nets.

To define s-nets, we begin with *vertices*, defined as the following types of expressions:

constant vertices	$c_1, c_2, \ldots$
variable vertices	$x_1, x_2, \ldots$
function vertices	$f_1^1, f_2^1, \ldots, f_1^2, f_2^2, \ldots$ (intuitively, the upper index indicates adicity)
predicate vertices	$P_1^1, P_2^1, \ldots, P_1^2, P_2^2, \ldots$
quantifier vertices	$\forall, \exists$
operator vertices	$\vee, \wedge, \sim, \supset$

In the following i, n are any positive integers.

term vertex: any constant or variable vertex, or any expression of form $(f_i^n t_1 \ldots t_n)$ where $t_1, \ldots, t_n$ are term vertices.

propositional vertex: any expression of form $(P_i^n\, t_1 \ldots t_n)$ where $t_1, \ldots, t_n$ are term vertices; or any expression of form $(Q\ x\ R)$ where Q is a quantifier vertex, x is a variable vertex, and R is a propositional vertex; or any expression of form $(\vee\, R\, S), (\wedge\, R\, S), (\sim R)$, or $(\supset R\, S)$, where R and S are propositional vertices.

We use labeled-edge terminology to describe the relation between an expression and its immediate (ordered) constituents. In a vertex V of form $(O\, V_1 \ldots V_n)$ (where necessarily O is a function, predicate, quantifier, or operator vertex), we say that there is a labelled directed edge $(V,\ O,\ OP)$ from vertex V to vertex O and a labelled directed edge (V, V_i, i) from vertex V to vertex V_i for $i = 1, \ldots, n$.

Finally, we define an *s-net* as a finite set of propositional vertices.

It is patently obvious that an s-net is just a set of FOPC formulas, in a slightly altered terminologocal guise. Yet an s-net also meets the minimal intuitive requirement for a semantic net, that of being a graphical structure (with vertices capable of representing things, concepts, and propositions, and edges supplying the "glue" that binds together the parts of functional concepts and propositions):

Theorem. The vertices and edges of an s-net form a directed, acyclic labelled graph without isolated vertices.

Proof. Briefly, this follows from the fact that the "proper subexpression" relation is irreflexive, and atomic vertices occur only as constituents of nonatomic ones. In more detail, a directed labelled graph is any triple of sets $\{V_1, ..., V_n\}$, $\{E_1, ..., E_m\}$, $\{L_1, ..., L_l\}$, where each E_i $(1 \leq i \leq m)$ is a triple (V_j, V_k, L) such that $1 \leq j, k \leq n$ and $L \in \{L_1, ..., L_l\}$. Obviously the vertex set, edge set, and label set $\{OP, 1, ..., N\}$ where N is the smallest integer ≥ 2 such that every function vertex f_i^n and predicate vertex P_i^n in the s-net has $n \leq N$, form a directed labelled graph.

An acyclic graph is one not containing a closed path, i.e., a subset of edges $\{(V_{j_1}, V_{j_2}, L_{l_1}), (V_{j_2}, V_{j_3}, L_{l_2}), ..., (V_{j_s}, V_{j_{s+1}}, L_{l_s})\}$, $(s \geq 1)$, such that $j_1 = j_{s+1}$. In an s-net, edges correspond to the "immediate subexpression" relation. Paths, therefore, correspond to the "proper subexpression" relation (not necessarily immediate). Hence if an s-net contained a closed path, it would contain an expression which has itself as a proper subexpression, an obvious impossibility.

An isolated vertex of a graph is one which does not occur in any edge. Clearly, an s-net has no isolated vertices, since each immediate subexpression of each non-atomic vertex is by definition not isolated (i.e., it occurs in the edge from the embedding vertex to it), and atomic vertices occur only as subexpressions of non-atomic ones (ultimately, of the propositional vertices comprising the s-net). □

Our sample isomorphism, then, is just the identity map from FOPC expressions to s-net vertices (or vice versa), these being one and the same thing. The claim about semantic equivalence (admitting the same valuations) is also trivially true as long as we agree to interpret s-net vertices just as we would logical expressions (which they are).

Of course, demonstrating isomorphism/equivalence for s-nets and FOPC doesn't demonstrate it for any of the many subspecies of nets and their putative logical counterparts. But at least it indicates how to proceed. A few further remarks are in order. s-nets are similar (in terms of representational properties) to the sorts of nets proposed by Shapiro (1971), Rumelhart et al. (1972), Schubert (1976), and many later schemes. In particular, there are explicit proposition nodes (vertices), and edge labels serve essentially to indicate argument order (though they may be chosen to remind us of uniformities across predicates: SUBJ, OBJ, and the like). Another popular type of net uses edges themselves as representations of binary predications, the labels being predicate symbols and thus freely interpretable (e.g., Winston 1970, Deliyanni & Kowalski 1979, Nilsson 1980). Here we naturally need a slightly different strategy for demonstrating the desired sort of isomorphism/equivalence. For instance, we might define a "propositional edge" as an expression of form $(P\ t_1 t_2)$ where P is an edge label (drawn from a set of labels interpretable as binary relations) and t_1, t_2 are term vertices. A (bare-bones) semantic net of this type would ultimately be defined as just a set of propositional edges – these entail the presence of the vertices they connect.

Also, many semantic net theorists have taken as their basic binary relations the "instance" relation between objects or events and their types, the "isa" relation between subtypes and supertypes, and case relations such as "agent-of", "object-of", "recipient-of", etc., tying participants in events to those events. This particular viewpoint appears to present no special obstacles to the sort of equivalence construction I have indicated. Even if its proponents were to deny that relationships such as (*agent-of John Kissing-event1*) can be formally evaluated in the manner of logical predications, it is hard to conceive of any alternative formal method of evaluation which would not, thereby, also provide an alternative formal semantics of binary predicate logic, and thus make the equivalence go through under that semantics.

Closely related to the notion of case relations is the frame-based notion of *slots* or *roles* associated with a concept, such as the parts of a thing or the participants in a situation. As Hayes (1979) points out, roles can be viewed as relations or Skolem functions, and as such are logically unproblematic. However, one representational feature of frames emphasized by Minsky (1975) is that they supply *default* characterictics for the object types they describe and their frame slots (roles). So, for instance, elephants are gray and have a rope-like tail by default, though specific exceptions (such as a white elephant with a deformed tail, or none at all) are permitted. (Much the same idea lies at the heart of *prototype* theory.) Are the network (or frame) representations of defaults beyond the pale of logic, and so a counterexample to the claimed representational equivalence?

Well again, the answer is that if we can find a formal way of making sense of defaults in role-structured nets (or frames), we'll also have a way of doing so for a linearized, "logical" representation of those defaults. There is nothing magical about drawing an arc labelled DEFAULT-V from the COLOR slot for the TYPICAL-ELEPHANT to GRAY, instead of writing down DEFAULT-V(COLOR(TYPICAL-ELEPHANT)) = GRAY. However, there *is* something slightly magical about drawing *conclusions* from either of these representations, without being able to say under what conditions "Elephants are typically gray" is *true* (which, as a matter of fact, it is). A good deal of effort is being devoted in AI and in linguistic semantics to this profound semantic puzzle; if and when this effort succeeds, we will also know what the correct net/logic mapping for default characterizations is.

In general, there is finicky work to be done in finding just the right logics to serve as isomorphic images of various network formalisms. For instance, the formalism in Schubert (1976) contains quantifier and operator scope conventions which generalize Skolem dependencies in a way that is not entirely trivial to map isomorphically into an ordinary logical format. More interestingly, it is easy to define network syntax so as to permit cycles, which, on a "subexpression" interpretation of network edges, cannot occur in an ordinary logical syntax. However, one can extend ordinary logical syntax in the following way. Introduce a set of *formula labels* p_1, p_2, ..., where the last element of each formula is to be a formula label (i.e., this last element comes immediately after the usual constituents of the formula). Thus, for instance, we write (*loves John Mary* p_1) rather than (*loves John Mary*). Labels of otherwise distinct formulas must also be distinct, and labels of otherwise identical formulas must also be identical. Furthermore, in forming any compound formula, the labels of the embedded formulas must be used in place of the formulas themselves. Thus we write

$(\wedge\ p_1\ p_3\ p_4)$, (*loves John Mary* p_1), (*loves Mary John* p_2), $(\sim\ p_2\ p_3)$

instead of

$(\wedge$ (*loves John Mary*) $(\sim$ (*loves Mary John*))).

Formulas are regarded as asserted only if their labels do not occur elsewhere. Thus, only p_4 above is regarded as asserted. It is easy to see how to formally interpret formulas of this new type, as long as there is no cyclic reference to formula labels. However, consider the following pair:

$(\supset$ *true* $p_1\ p_2)$, $(\sim\ p_1\ p_1)$

Here p_2 is asserted; it says that truth implies p_1 (assuming *true* is an atomic formula interpreted as truth), and so asserts p_1. But p_1 says that p_1 is not the case, and so denies itself. Thus, we have a paradox.

This extended logic models potentially cyclic semantic nets in an obvious way, and reflects the potential for paradox in such nets. Again, however, this does not create any problem for isomorphism between nets and logic. Rather, it creates a problem for the semantics of *both* nets and the corresponding logic. Any formal semantics successfully addressing this problem in one formalism will immediately address it in the other, as well. (The non-well founded set theory of Aczel, 1986, or the truth-revision theory of Gupta, 1987, may perhaps provide a basis for a solution.)

3 Computational Properties, and a State-of-the-Art System

The clustering of properties around concepts, and the incorporation of inheritance hierarchies, have often been held up as the most significant features of semantic nets; and so they are. The trouble, from a terminological standpoint, is that neither of these features is at this stage still distinctive of semantic nets. On

THIS

PAGE

IS

MISSING

IN

ORIGINAL

DOCUMENT

9

The same point applies, e.g., to property inheritance. A logical view of the inference that

(*color Clyde gray*)

for instance, based on

(*elephant Clyde*), ($\forall x$($\supset$ (*elephant x*) (*color x gray*)))

would again involve "nearby" s-net vertices, and might well be facilitated by the previous sort of hash-indexing under (predicate, argument) pairs such as (*elephant, Clyde*), (*elephant*, \$), (*color*, \$), and (*color, gray*), with an inference strategy that pays special attention to "type predications" such as (*elephant Clyde*), accessing information about the type via index (*elephant*, \$). ("\$" is here used as a uniform variable token, for hash purposes.) This strategy can be improved upon (e.g., deHaan & Schubert 1986), but whether one regards the improved strategy as a logical inference strategy or a semantic net strategy is purely a matter of terminological preference.

These issues have been very much on my mind in the choice of terminology for describing ECOLOGIC, an inferential knowledge base recently implemented at the University of Alberta (see Schubert & Hwang 1989, 1990). The system is intended to support a general narrative understanding and question answering system. (ECO derives from English COnversation, and LOGIC from the system's logical soul.) The logic is intensional, probabilistic (allowing unreliable generalizations and degrees of belief), and rather close to English surface structure. ECOLOGIC incorporates many features of its predecessor, ECONET, which we still hesitantly termed a semantic net: a concept-centered, topic-oriented retrieval mechanism, type and topic hierarchies, and a general goal-directed inference mechanism aided by several "specialists" that shortcircuit temporal, taxonomic, set and number inferences (Miller & Schubert 1988). In addition it performs input-driven inference, working out the probable consequeces and explanations of logical-form inputs. Both systems are able to infer answers to questions such as "Did the wolf eat a person?", and "Does Grandmother live in a shoe?" based on a simplified, logically encoded version of Little Red Riding Hood. The new system, however, also anticipates that Little Red Riding Hood is probably in danger when she meets the wolf, and makes many other inferences spontaneously which could previously be obtained only in goal-directed fashion.

Though the "semantic competence" of our most recent system thus improves over our previous ones, we now generally avoid the term "semantic net". Already in the design of ECONET, we ceased drawing our propositions as graphs; the propositions in the new, enriched syntax (admitting restricted quantification, lambda abstraction, and nominalization, among other things) would be even less readable when depicted by swirling lines rather than formulas. This seemingly insignificant fact greatly diminished our disposition to think in semantic net terms. Perhaps the last straw was the fact that input-driven inference can be quite effectively implemented by hash-table methods of the sort mentioned above – an access method not particularly indebted to semantic net theory.

4 What's Left?

To say that semantic nets, in the broadest sense, lack distinctive representational and computational properties is not to say that the term should be expunged from the AI lexicon. (Far be it from me to advocate lexical depletion.) There *are* some promising, and quite familiar, directions for useful redeployment of the term.

One possibility, which I discussed in (Schubert 1976), is to reserve it for the graphical *depictions* of propositional information we find helpful in conceiving or explaining certain inferential processes. (As such, they would be viewed as analogous to Venn diagrams in set theory.) However, I doubt that many people are ready to think of semantic nets as mere pictures, however strongly they may feel about the advantages of node-and-link diagrams or nested boxes. That is why I set aside the diagrammatic aspect from the outset. A better possibility is to reserve the term for propositional representations which, like Quillian's original networks, make essential use of *spreading activation* or similar propagation processes for inference or understanding. This use may be compatible with some of the current work in neural nets (inasmuch as this work also depends upon concept-to-concept signal propagation). A third is to reserve it for special-purpose graphical structures and their associated inference mechanisms, such as taxonomic hierarchies, parts hierarchies, or time graphs. These have, of course, proved very useful as enhancements to general propositional inference systems.

With regard to this last possibility, I need to emphasize two things. First, a representation is graphical in the sense I intend only if it uses graph-theoretic notions in an essential, nontrivial way. The mere possibility of depicting a representation in node-link form does not, in itself, demonstrate that the representation is intrinsically graphical; nor does the use of an inference strategy which prefers to combine closely linked facts necessarily show this; whereas systematic reliance on formally defined distance metrics or topological properties (connectivity, cycles, classes of paths, etc.) for inference (deduction, analogy, associative retrieval, etc.) may indeed do so.

Secondly, I need to emphasize the distinction between graph-based auxiliary representations on the one hand, and the larger, more general inference systems which *incorporate* them on the other. If we were to call a representation a semantic net merely because it *incorporates* a taxonomic subsystem (say, to support property inheritance), we would revert precisely to the practice I am decrying! The point is that a taxonomic (or temporal, etc.) reasoner which makes *essential* use of *nontrivial* graph-theoretic properties is, on account of that, a member of an important and objectively distinguishable species of knowledge representation. A system which merely incorporates some such special reasoner is, nowadays, a system like any other.

Acknowledgements

I am grateful for the helpful and provocative comments (in the best sense of the word) on the manuscript by Roger Hartley, Alfred Kobsa, John Sowa and (at the Catalina workshop) Bill Woods. The research was supported by operating grant A8818 of the Natural Sciences and Engineering Research Council of Canada, by the Boeing Company under Purchase Contracts W-278258 and W-288104, and ONR/DARPA research contract no. N00014-82-K-0193.

References

Aczel, P. (1986). Lectures on non-well founded sets, CSLI, Stanford, CA.

Coles, L.S. (1972). Techniques for information retrieval using an inferential question answering system with natural language input, SRI Tech. Rep., SRI International, Menlo Park, CA.

deHaan, J., and Schubert, L.K. (1986). Inference in a topically organized semantic net, *Proc. AAAI-86*, pp. 334-8.

Deliyanni, A., and Kowalski, R. (1979). Logic and semantic networks, *CACM 22*, 184-192.

Fahlman, S.E. (1974). A planning system for robot construction tasks, *Artificial Intelligence 5*, pp. 1-49.

Gupta, A. (1987). The meaning of truth. In E. LePore (ed.), *New Directions in Semantics*, Academic Press, New York, pp. 453-480.

Hayes, P.J. (1979). The logic of frames. In D. Metzing (ed.), *Frame Conceptions and Text Understanding*, de Gruyter, Berlin, pp. 46-61. Also in B.L. Webber and N.J. Nilsson (eds.), *Readings in Artificial Intelligence*, Morgan Kaufmann, Los Altos, 1981, pp. 451-8.

Hendrix, G.G. (1973). Language processing via canonical verbs and semantic models, *Adv. Papers of IJCAI-3*, pp. 262-9.

Miller, S.A., and Schubert, L.K. (1988). Using specialists to accelerate general reasoning, *Proc. of AAAI-88*, Aug. 21-26, St. Paul, MN, pp. 161-5.

Minsky, M. (1975). A framework for representing knowledge. In P.H. Winston (ed.), *The Psychology of Computer Vision*, McGraw-Hill, New York, pp. 211-277.

Nilsson, N.J. (1980). *Principles of Artificial Intelligence*, Tioga Publ. Co., Palo Alto, CA.

Rumelhart, D., Lindsay, P., and Norman, D. (1972). A process model for long term memory, in E. Tulving and W. Donaldson (eds.), *Organization of Memory*, Academic Press, New York, pp. 198-221.

Schubert, L.K. (1976). Extending the expressive power of semantic nets, *Artificial Intelligence* 7, pp. 163-198.

Schubert, L.K., and Hwang, C.-H. (1989). An episodic knowledge representation for narrative texts. *Proc. of the 1st Int. Conf. on Principles of Knowledge Representation and Reasoning* (KR'89), May 15-18, 1989, Toronto, Canada, pp. 444-458.

Schubert, L.K., and Hwang, C.-H. (1990). An episodic knowledge representation for narrative texts, Tech. Rep. 345, Dept. of Computer Science, U. of Rochester. (This substantially elaborates the KR'89 version.)

Shapiro, S.C. (1971). A net structure for semantic information storage, deduction, and retrieval. *Adv. Papers of IJCAI-2*, pp. 512- 523.

Winston, P. (1970). Learning structural descriptions from examples, Ph.D. Thesis, MIT, MAC-TR-76, Cambridge, MA.